Department of Electrical Engineering
School of Science and Engineering

EE514/CS535 Machine Learning

HOMEWORK 3 – SOLUTIONS

Due Date: 23:55, Sunday, April 25, 2021 (Submit online on LMS)
Format: 5 problems, for a total of 80 marks
Contribution to Final Percentage: 2.5%
Instructions:

- Each student must submit his/her own hand-written assignment, scanned in a single PDF document.
- You are allowed to collaborate with your peers but copying your colleague's solution is strictly prohibited. Anybody found guilty would be subjected to disciplinary action in accordance with the university rules and regulations.
- Note: Vectors are written in lowercase and bold in the homework, for your written submission kindly use an underline instead. In addition, use capital letters for matrices and lowercase for scalars.

Problem 1 (10 marks)

Perceptron - Consider a single perceptron classifier that receives binary input $\boldsymbol{x} \in \{0,1\}^d$, shown in Figure 1.

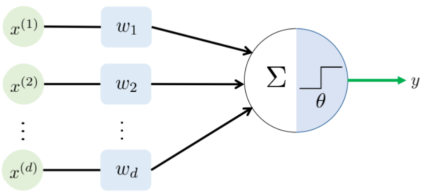

Figure 1: Perceptron Classifier

Where:

$$y = \begin{cases} 1 & \sum_{i=1}^{d} w_i x^{(i)} \geq \theta \\ 0 & \sum_{i=1}^{d} w_i x^{(i)} < \theta \end{cases}$$

(a) [**4 marks**] Given $d = 2$, threshold $\theta = -3$. Provide the values of weights, w_1 and w_2, if we were to construct a NAND function.

(b) [**6 marks**] Now repeat the same problem with $d = 3$ and NOR function.

Solution:

(a) Possible solution could be $w_1, w_2 = (-2, -2)$

(b) Possible solution could be $w_1, w_2, w_3 = (-4, -4, -4)$

Problem 2 (20 marks)

Hard Margin SVM - Consider the training data points given in Figure 2. Positve class has label '+1' and are shown as red circles, while negative data points have label '-1' and are shown as blue triangles.

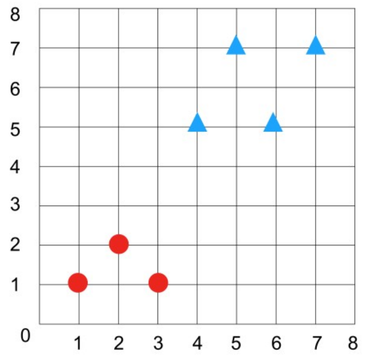

Figure 2: Training Points for SVM

(a) **[4 marks]** In the case that we use a hard-margin SVM classifier, write down the support vectors. Show your working and diagrams.

(b) **[6 marks]** Now suppose we add another negative class data point at (5,1), write down the new support vectors. Show your working and diagrams.

(c) **[5 marks]** Consider the general scenario of adding new training data points in a SVM classification problem. Will our objective function $\|w\|$ decrease? Increase? Or stay the same? Justify each case.

(d) **[5 marks]** Explain why we require **at least** one data point on each side of the decision boundary to converge to a solution in SVM hard margin classification.

Solution:

(a) Support vectors are: (2,2) and (4,5)

(b) Support vectors are: (3,1), (4,5) and (5,1)

(c) The objective function can increase or stay the same, however, it can not decrease.
If data points are added outside the margins in their respective classes, then our decision boundary remains the same. However, adding a data point in the margin area results in a thinner margin and a larger $\|w\|$. There is no way to improve upon the old objective function, i.e., maximize the margin further.

(d) If we have only one data type, we can always improve upon our objective function by scaling w to make it smaller. That is, we can keep maximizing our margin since there is nothing limiting us on the other end.

Problem 3 (10 marks)

Perceptron and SVM - In this problem we will look at a comparison of Perceptron and SVM and how to counter their limitations.

(a) [**4 marks**] Consider the single dimension classification problem given in Figure 3. Provide a range of possible decision boundaries for perceptron and hard-margin SVM.

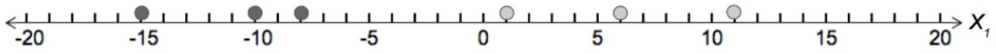

Figure 3: Single dimension training data

(b) [**6 marks**] Now consider the training data in Figure 4.

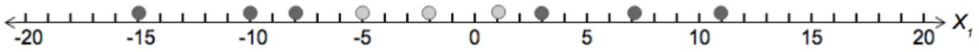

Figure 4: Single dimension training data

Clearly the problem is no longer linearly separable. However we can convert it into a linearly separable problem by introducing a new feature given as:

$$x_2 = \begin{cases} 3 & x_1 \le a \\ -3 & x_1 > a \end{cases}$$

Provide a value of a that can accomplish this task. Show that separation is achieved by plotting a decision boundary (give equation of line). You may use online graphing softwares such as Desmos to help visualise and plot the two dimensional data.

Solution:

(a) Perceptron: [-8, 1]
Hard-margin SVM: -3.5

(b) We can choose a in range [-8, -5) to allow for linear separation. See Figure given below:

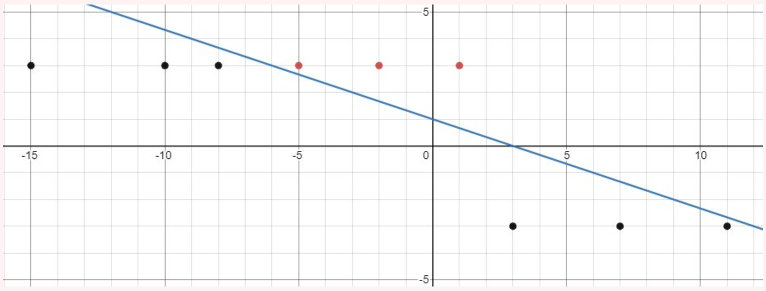

Figure 5: Decision boundary in two dimensions

Equation of line shown is: $x + 3y = 3$.

Problem 4 (30 marks)

Backpropgation - In this problem we will use a neural network to implement a self-driving car. You are provided images of the steering wheel, which are processed into grayscale 128 × 128 pixel images, these act as the input to the network. There are two labels of each image, the steering wheel angle and speed in kilometers per hour.

Your neural network has a hidden layer with 1024 units. The activation function used in hidden layer is the ReLU function. (There is no activation function at the output).

(a) [**5 marks**] Draw and label the neural network.

(b) [**5 marks**] Calculate the **total** number of weights in the neural network. (Be mindful of the bias).
The loss function chosen is given by: $J = \frac{1}{2}|\boldsymbol{y} - \hat{\boldsymbol{y}}|$, where $\boldsymbol{y}$ is the true training label vector, and $\hat{\boldsymbol{y}}$ is the predicted output vector. You are given the following transformations:

- $\boldsymbol{x}$ is the flattened image vector with bias appended
- $\boldsymbol{z}$ is the vector of values at hidden layer, before activation function.
- $W^{[1]}$ is the weight matrix mapping from the input layer to the hidden layer, i.e., $\boldsymbol{z} = W^{[1]}\boldsymbol{x}$
- $g(.)$ is the ReLU activation function.
- $\boldsymbol{a}$ is the vector of values at hidden layer, after passing through activation function, i.e., $\boldsymbol{a} = g(\boldsymbol{z})$
- $W^{[2]}$ is the weight matrix mapping from hidden layer to the output layer, i.e., $\hat{\boldsymbol{y}} = W^{[2]}\boldsymbol{a}$

(c) [**5 marks**] Derive $\frac{\partial J}{\partial W_{ij}^{[2]}}$

(d) [**5 marks**] Now write $\frac{\partial J}{\partial W^{[2]}}$, i.e., use vector products.

(e) [**10 marks**] Using your result from previous parts, derive $\frac{\partial J}{\partial W_{ij}^{[1]}}$

Solution:

(a) The neural network is shown below:

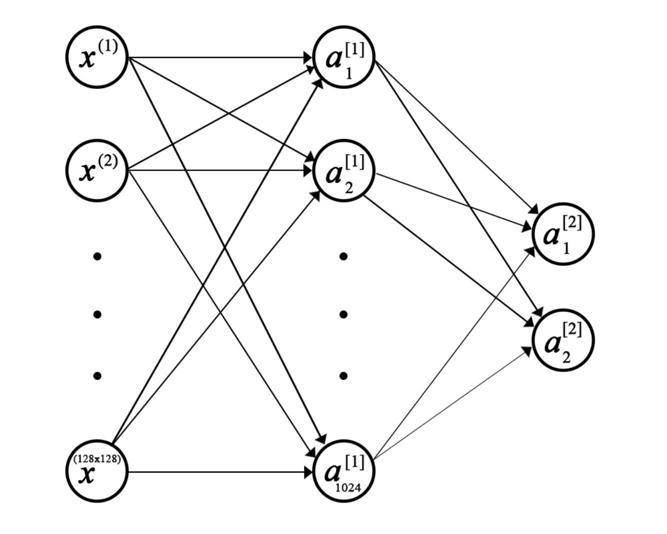

Figure 6: Neural network for problem 4

(b) $\big(d \times 1024 + 1024\big) + \big((1024 \times 2) + 2\big)$, where $d = 128 \times 128$.

(c)

$$\begin{aligned}\frac{\partial J}{\partial W_{ij}^{[2]}} &= (\hat{\boldsymbol{y}} - \boldsymbol{y})^T \frac{\partial \hat{\boldsymbol{y}}}{\partial W_{ij}^{[2]}} \\ &= (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)\boldsymbol{a}_j\end{aligned}$$

(d)

$$\frac{\partial J}{\partial W^{[2]}} = (\hat{\boldsymbol{y}} - \boldsymbol{y})^T \boldsymbol{a}$$

(e)

$$\begin{aligned}\frac{\partial J}{\partial W_{ij}^{[1]}} &= (\hat{\boldsymbol{y}} - \boldsymbol{y})^T \frac{\partial \hat{\boldsymbol{y}}}{\partial W_{ij}^{[1]}} \\ &= (\hat{\boldsymbol{y}} - \boldsymbol{y}) W^{[2]} \frac{\partial \boldsymbol{a}}{\partial W_{ij}^{[1]}} \\ &= (\hat{\boldsymbol{y}} - \boldsymbol{y}) W^{[2]} [0, \ldots, g^{'}(z_i) x_j, \ldots, 0]^T \\ &= ((\hat{\boldsymbol{y}} - \boldsymbol{y})^T W^{[2]})_i g^{'}(z_i) x_j\end{aligned}$$

Problem 5 (10 marks)

k-Means Clustering - In this question we will dry-run the k-means clustering algorithm on a two dimensional data-set. The data-set is provided in Table 1.

#	Data Point
1	(2, 5)
2	(1, 7)
3	(1, 3)
4	(3, 2)
5	(5, 2)
6	(2, 7)
7	(3, 5)
8	(2, 2)

Table 1: Data points for unsupervised learning

Run the k-means algorithm for $k = 3$ on this data-set until convergence. Take data points 3, 6 and 7 as your initial centroids for k = 1, 2, and 3 respectively. Use Euclidean distance where required. Classify each data point with it's corresponding k-number.

Solution: First iteration:

#	Data Point	Distance to			Cluster Assigned
		μ_1 (1,3)	μ_2 (2,7)	μ_3 (3,5)	
1	(2, 5)	2.24	2	1	3
2	(1, 7)	4	1	2.83	2
3	(1, 3)	0	4.12	2.83	1
4	(3, 2)	2.24	5.10	3	1
5	(5, 2)	4.12	5.83	3.61	3
6	(2, 7)	4.12	0	2.24	2
7	(3, 5)	2.83	2.24	0	3
8	(2, 2)	1.41	5	3.16	1

Table 2: Iteration 1

Updated cluster means are: $\mu_1 = (2, 2.33), \mu_2 = (1.5, 7), \mu_3 = (3.33, 4)$
Second iteration:

#	Data Point	Distance to			Cluster Assigned
		μ_1 (2,2.33)	μ_2 (1.5,7)	μ_3 (3.33,4)	
1	(2, 5)	2.67	2.06	1.66	3
2	(1, 7)	4.77	0.5	3.80	2
3	(1, 3)	1.20	4.03	2.53	1
4	(3, 2)	1.05	5.22	3.60	1
5	(5, 2)	3.01	6.10	2.60	3
6	(2, 7)	4.67	0.5	3.28	2
7	(3, 5)	2.85	2.5	1.05	3
8	(2, 2)	0.33	5.02	2.40	1

Table 3: Iteration 2

Means of all clusters remain the same, therefore we have converged.

— End of Homework —